

日 本 国 特 許 庁

PATENT OFFICE  
JAPANESE GOVERNMENT

JC688 U.S. PRO  
09/742657  
12/21/88

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。 #4

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出 願 年 月 日

Date of Application:

1999年12月28日

出 願 番 号

Application Number:

平成11年特許願第375718号

出 願 人

Applicant(s):

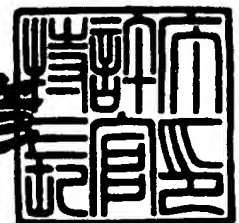
日本アイ・ピー・エム株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2000年 5月12日

特許庁長官  
Commissioner,  
Patent Office

近 藤 隆 彦



出証番号 出証特2000-3034154

【書類名】 特許願

【整理番号】 JA999204

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 12/00

【発明者】

【住所又は居所】 東京都中央区日本橋箱崎町 1 9 番地 1 日本アイ・ピー  
・エム株式会社 箱崎事業所内

【氏名】 西川 英徳

【特許出願人】

【識別番号】 592073101

【住所又は居所】 東京都港区六本木 3 丁目 2 番 1 2 号

【氏名又は名称】 日本アイ・ピー・エム株式会社

【代理人】

【識別番号】 100086243

【弁理士】

【氏名又は名称】 坂口 博

【連絡先】 0 4 6 2 - 7 3 - 3 3 1 8、3 3 2 5、3 4 3 1

【選任した代理人】

【識別番号】 100091568

【弁理士】

【氏名又は名称】 市位 嘉宏

【手数料の表示】

【予納台帳番号】 029193

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9306921

【包括委任状番号】 9306922

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 階層リンク・テーブルを備えたデータベース・システム

【特許請求の範囲】

【請求項 1】

一定の業務を実行するための業務処理プログラムの処理で用いられるデータを格納し管理するためのデータベース・システムであって、  
上記業務処理プログラムの処理で用いられるデータをノード・データとしてデータレコードに格納する階層ノード・データベースと、  
上記業務処理プログラムごとに対応して設けられ、上記階層ノード・データベースに格納されたデータ・ノード間の階層構造を定義する関連付け情報をデータレコードのデータ項目として保持する階層リンク・テーブルと、  
を具備するデータベース・システム。

【請求項 2】

上記階層リンク・テーブルが、各データレコードの有効期間を定義する有効期間情報をそれぞれのデータレコードのデータ項目として保持する、請求項 1 に記載のデータベース・システム。

【請求項 3】

上記階層ノード・データベースが、各データレコードの有効期間を定義する有効期間情報をそれぞれのデータ・フィールドのデータ項目として保持する、請求項 1 または 2 に記載のデータベース・システム。

【請求項 4】

上記階層ノード・データベースの各データレコードが、データサイズが一定であるデータ項目のみを格納する固定長カラムと、データサイズが可変のデータを格納する可変長カラムを有する、請求項 1 ないし 3 のいずれか 1 に記載のデータベース・システム。

【請求項 5】

一定の時間間隔で業務処理が実行される業務処理プログラムの実行のタイミングを定義するサイクル情報を有するサイクル制御表をさらに具備する、請求項 1 ないし 4 のいずれか 1 に記載のデータベース・システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、一定の業務処理を行なうアプリケーションプログラム（業務処理プログラム）がアクセスするデータベース・システムに係り、特に業務内容や関連データの追加・変更に対しても柔軟に対処できるデータベース・システムに関する。

【0002】

【従来の技術】

一定の業務処理を行なうアプリケーションプログラム（業務処理プログラム）が扱うデータはその量が大きいため、通常データベース・システムに格納されて管理される。このデータベース・システムは、典型的には概ね次のようにして構築される。まず、すべての業務処理プログラムに必要なデータベース（エンティティともいう）と当該データベース内のデータ項目（各エンティティの属性項目ともいう）を洗い出す。そして、想定されるすべての業務処理プログラムが効率的な処理を行なえるように、データベース間の関係（親子関係等の階層構造）およびデータベース内のデータ項目間の関係（これらの関係のことをリレーションシップともいう）を定義または決定する。決定されたこれら関係は、ER図（エンティティ・リレーションシップ・ダイアグラム）として表現することができる。このER図は、データベース管理システム（DBMS）の機能を用いて、実際のデータベースとして定義される。例えば、リレーショナル・データベースの場合には、SQLのような言語を用いてデータベースが定義される。その結果、各データベースにおいては、各データの実際の値や内容等（インスタンス）を格納するレコード中（キーエリア、カラム等）に、関連するデータベースのレコード識別子を示す関連付け情報（ポインタ等）が生成される。このような関連付け情報（ポインタ等）は、実際には、通常の業務処理の一環として業務処理プログラムが作成することが多い。

【0003】

業務の処理要件の変更や新しい業務の追加が行なわれたり、データベースが追加

／変更されたりすると、データベース間の関係／階層構造が修正（例えば、親子関係の解消／追加／逆転等）されることになる。この場合、あらためて先に述べたようなデータベース間の関係付け作業を行って、変更後の階層構造をデータベース管理システム（DBMS）の機能を用いてデータベース中に作りこむという変更作業が必要になる。また、既存の各データベース内のレコード中（カラム）に存在するポインタ等の関連付け情報も変更しなければならない。具体的には、変更のためのプログラムを作成してデータベースに変更を加える。

#### 【0004】

このような変更作業が行なわれると、変更されたデータベースを使用（データ項目を読み出して処理したり、書き込み／更新したりすることをいう）しているすべての業務処理プログラムに対して、データベースの階層構造の変更に応じた内容の修正か、少なくともその影響が無いことの確認のテストを行なう必要が生じる。データベース間の関係すなわちデータベースの階層構造には変更がないが、データベース内のレコードとこれに関係付けられた別のデータベース内のレコードとの間の関連付けが変更されるといったこと（例えば、顧客Aがある電話番号を別の人に譲渡した場合等のように、いわゆるインスタンス・レベルで親子関係の変更／所属変更等が発生すること）は、通常の業務処理の一環として発生することであり、業務処理プログラムがかかる変更処理を実施する。

#### 【0005】

ある特定の業務処理プログラムから見た場合、このプログラム自身が使用するデータベース群の間にはその処理上必要な関係や階層構造が存在するが、この関係／階層構造は、全データベース間の関係の一部を構成している（部分集合となっている）ので、当該業務処理プログラム用の関係／階層構造を個別に作成しておく必要は必ずしもない。すなわち、全データベース間の関係を示すポインタ等の関連付け情報をたどっていけば、必要なデータベースにはたどり着くことができる。しかしながら、処理効率等を考慮して、通常は当該業務処理プログラムに対応する関係／階層構造を、ビューと呼ばれるデータベース管理システム（DBMS）の機能を用いて作成しておく。業務処理プログラムは、このように別途作成されたビュー等を参照することにより、該当するデータベース群にアクセスする

ことができる。

【0006】

また、別の観点からこれらの業務処理に用いられるオペレーショナルなデータベースに着目すると、各データベース中の個々のレコードには、通常「最新」のデータの実際の値／内容（インスタンス）が格納されており、この「最新」のデータのみが利用可能になっている。ここで「最新」とは、前日の夜間の処理終了時点での値（夜間バッチ処理結果等）であったり、取引処理（トランザクション）が発生する毎に追加／更新された最新状態（リアルタイム処理結果）であったりする。例えば、業務処理プログラムが一定の期間単位で処理を行うような場合、例えば、1ヶ月間に格納されたレコードをその月の月末にまとめて集計／計算処理するような場合等に、当該業務処理プログラムは、当月末にデータベースにアクセスして、その中に格納されている最新の全データレコードに基づき一定の処理を行なう。しかしながら、その月の途中のある時点でデータベース間またはデータレコード間の関係が変更される場合がある（例えば、顧客Aがある電話番号を別の人に譲渡した場合等のように、いわゆるインスタンス・レベルで親子関係の変更／所属変更等が発生する場合）。この場合、従来の関連付け情報（ポインタ等）には、その関係がいつからいつまで有効であったかを示すようなデータは保持されていない。その結果、その変更が発生した時点で、一旦、最新のデータおよび関連付け情報に基づいて当該業務処理プログラムを実行させ、その変更時点以降につき月末になった時点で、再度最新のすなわち変更後のデータおよび関連付け情報に基づいてプログラムを実行させる必要が生じる。

【0007】

【発明が解決しようとする課題】

しかしながら、上述のようなデータベースの構築の手法において、想定されるすべての業務処理プログラムが必要とするすべてのデータベース間の関係を事前に洗い出して定義または決定することは極めて困難であり、かつ非常に時間がかかる。このため、実際には、ある程度のところで妥協して定義・決定を行うので、後にデータベース間の関係の変更作業が頻繁に発生し、これに付随して多くの業務処理プログラムの変更やテスト作業が生じ、多大な労力と時間を消費すること

になる。上述のようなビューと呼ばれるデータベース管理システム（DBMS）の機能を用いて作成された階層構造は、あくまで全体の階層構造の部分集合であり、データベースの実体そのものではない。よって、これを単独に変更することはできず、同様の問題が生じることに変わりはない。

## 【 0 0 0 8 】

このことは、新しい業務の追加や既存の業務内容の変更に対して、柔軟かつ迅速に対応できるデータベース・システムの提供ができないことを意味する。すなわち、新しいデータベース間の関係／階層構造を持つような新しい業務上の要件が発生しても（例えば、競争優位のために従来にないデータベース間の親子関係／グルーピング関係を持つ新しい商品やサービスを提供したいと思っても）、迅速にデータベース・システムを更新することはできない。また、ある業務処理プログラムにとって必要なデータベース間の親子関係と別の業務処理プログラムにとって必要な親子関係が逆転しているような場合、あるいは業務上の要件が変更されデータベース間の親子関係を逆にしたい場合には、そもそもデータベースの階層構造を定義できないこともあり、そのような業務要件を実現できないこともある。また、ある業務処理プログラムが、基準となる特定のデータベース（親エンティティ）とその下に関係付けられるデータベース（子のエンティティ）群を処理するように設計されていたのに、新しい要件として当該基準データベース（親エンティティ）より上位（親の親の位置）に別のエンティティを追加し、これを基準としてデータベース群を処理しなければならなくなった場合（例えば、「会社」を基準とする単位で請求や割引処理をするような業務処理プログラムがあったとして、「会社」より上位のエンティティとして「持株会社」を追加して、これを基準とする単位で処理しなければならなくなったような場合）、データベース、その関連付け情報、関与する業務処理プログラムの大幅な変更が必要となり、実質的には、その実現をあきらめることも多い。さらに、上記の例で、二つの基準となるデータベース（エンティティ）のデータレコード（インスタンス）が合体して一つになるような場合（例えば、二つの会社が合併して一つになったような場合）にも同様の問題が生じる。

## 【 0 0 0 9 】



さらに別の観点からの問題として、一定の期間単位で処理を行なう業務処理プログラムにおいては、その期間の途中でデータベースのデータレコードの変更やデータベース間の関係の変更が生じた場合、上述のように煩雑なプログラムの実行処理が要求される点が挙げられる。これは、通常、業務処理に用いられるデータベースが「最新」という一時点に対応するデータ内容を維持するのみであるため、そのデータベースだけにアクセスしても、過去に溯っての処理はできないことに起因する。このような場合、データベース等の変更が発生する都度プログラムを実行したり、または過去の記録を保存するファイル（例えば、ログ・ファイル等）を読み出してきて特別の処理を実行したりしなければならず、業務処理プログラム等にそれ用の特別の機能を作りこんでおく必要がある。これは、業務処理プログラムの複雑化や膨大化、品質の悪化、開発・維持コストの増大につながる。

#### 【0010】

また、ビューと呼ばれるデータベース管理システム（DBMS）の機能を用いて作成された階層構造は、上述のようにデータベースの実体そのものではないため、業務上必要となるこれらの有効期間情報を任意に設定することはできず、上記で述べたような課題の解決にはならない。

#### 【0011】

##### 【発明を解決するための手段】

本発明は、上記のような問題、すなわち業務内容やデータベースの内容・関係の変更に対して既存のデータベース・システムが柔軟かつ迅速に対応できないという問題を解決することをその目的とする。そのため、本発明は、データベース（エンティティ）間の関係付けを、データベースや業務処理プログラムの開発／維持に大きな負担をかけることなく柔軟に設定・変更できるデータベースの管理方式、ならびに業務処理プログラムによりデータベースを処理する時期と、データベース間の関係付けを追加・変更する時期またはデータレコード（インスタンス）を追加・変更する時期とを独立に扱い、互いの依存度を高めて等時間軸（時間の流れ）に関する柔軟性をもたらすデータベース管理方式を提供する。これによって、変更に対して柔軟なデータベース・システムを実現することが可能となる。

。

## 【0012】

具体的には、本発明においては、まず処理対象となる各データベース中（およびそのデータレコード内）にデータベース（エンティティ）間の関連付け情報（ポインタ等）を原則として持たせない。このようにして構成するデータベースを「階層ノード・データベース」という。一方、これとは別個にデータベース（エンティティ）間の関連付け情報を主として有するテーブル（データベースの一種）を各業務処理プログラムに対応して作成する。このようにして構成されたテーブルを「階層リンク・テーブル」という。各業務処理プログラムは、対応する階層リンク・テーブルを参照することにより、自分が必要とする階層ノード・データベースにアクセスすることができる。

（ただし、階層リンク・テーブルが何らかの原因で壊れても別ルートを辿って復旧ができるようにといった処理性能上の理由等から、階層ノード・データベースの中に一部の関連付け情報をもたせる方式を採用してもよい。）

## 【0013】

この本発明のデータベース管理方式によれば、想定されるすべての業務処理プログラムが必要とするデータベース等を、事前にすべて洗い出して定義／決定する必要はなく、当面想定される必要なデータベースのみを各業務処理プログラムの要件毎に階層リンク・テーブルに設定しておけばよい。新しい要件に基づく業務処理プログラムが追加されたときには、それに対応した修正を階層リンク・テーブルに反映するだけでよい。また、ある業務処理プログラムにとって必要なデータベース間の親子関係と別の業務処理プログラムにとって必要な親子関係が逆転する場合や、新たな業務上の要件が変わって親子関係を逆にしたい場合にも、データそのものを格納する階層ノード・データベースには手を加える必要はなく、それぞれの業務処理プログラムに対応する階層リンク・テーブルにのみ必要な親子関係／階層構造を定義すればよい。また、基準となる特定のデータベース（エンティティ）の追加・修正が生じた場合（上述の「持株会社」の例）や二つの基準となるデータベース（エンティティ）のデータレコード（インスタンス）が合体する場合（上述の会社の合併の例）も同様である。

## 【0014】

階層リンク・テーブルの各データレコードは、関連付け情報（ポインタ等）と必要な場合には後述する有効期間情報等が格納されているのみで比較的短いレコードであるため、階層リンク・テーブルの変更作業自体は、トラブルに備えて各種バックアップ等を取るといった作業を含めても、短時間で容易に実行できる。また、対応する業務処理プログラムは、一個か極めて少ない数でその影響範囲が限定されているため、変更後の確認テストも比較的少ない手間で行なうことができる。

## 【0015】

さらに、本発明は、上述の階層リンク・テーブルの各データレコードに、そのデータレコードがポインタ情報として保持している階層ノード・データベース（データレコード）間の親子関係／階層構造が、対応する業務処理プログラムからみて有効である期間、すなわち有効になる日時（有効開始日）と有効でなくなる日時（有効終了日）を設定することもできる。このように設定される有効開始日と有効終了日をあわせて有効期間情報という。この有効期間情報の付加により、各業務処理プログラムが過去に溯って処理を実行することができる。

## 【0016】

階層ノード・データベースに有効期間情報を設定するか否かは任意である。階層ノード・データベースは、一般に、各種業務処理プログラムから共通にアクセスされるので、アクセスしてくるすべての業務処理プログラムにとって有効な日時をセットするか、主要な業務処理プログラムにとって有効な日時を複数個セットするか、一切セットしないか、のいずれをも選択できる。逆に、特定の業務処理プログラムに対してのみ有効期間を設定したい場合には、階層リンク・テーブル側の有効期間情報を調整すればよく、階層ノード・データベース側でかかる情報を保持する必要はない。

## 【0017】

以上まとめると、本発明は、一定の業務を実行するための業務処理プログラムの処理で用いられるデータをノード・データとしてデータレコードに格納する階層ノード・データベースと、各業務処理プログラムごとに対応して設けられ、上記

階層ノード・データベースに格納されたデータ・ノード間の階層構造を定義する関連付け情報をデータレコードのデータ項目として保持する階層リンク・テーブルとを具備するデータベース・システムによって実現される。この階層リンク・テーブルは、各データレコードの有効期間を定義する有効期間情報をそれぞれのデータレコードのデータ項目として保持することができる。また任意ではあるが、階層ノード・データベースがこのような有効期間情報を保持してもよい。

【0018】

【発明の実施の形態】

以下、本発明の実施の形態について説明するが、ここでは説明の便宜のために電話料金等の費用の計算処理（割引処理）システムに適用した実施例を用いる。しかしながら、本発明はかかる実施例に限定されるものではなく、変更に対し柔軟なデータベース・システムの構築を意図するものであれば他の実施例にも適用できることに留意されたい。

【0019】

図1は、本発明を構成するデータ処理システムを説明するブロック図である。特定の業務処理を行なうためのアプリケーションプログラムである業務処理プログラム100は、データベース管理システム200を通じて、その処理に必要なデータをデータベース・システム300から取得する。本発明のデータベース・システム300には、各業務処理プログラム100に対応する形で階層リンク・テーブル310が設けられている。この階層リンク・テーブル310は、データベース（エンティティ）間の関連付け情報（ポインタ等）を保持するテーブルであり、対応する業務処理プログラム100の処理に必要なデータを有する階層ノード・データベース320の階層構造にかんする情報を保持する。階層ノード・データベース320は、原則として、かかる関連付け情報をもたないデータ本体を保持するデータベースである。なお、図示していないが、後述するサイクル制御表のようなテーブル情報もデータベース310に格納されている。

【0020】

次に、本発明の階層リンク・テーブルによるデータベース管理方式を従来の方式と比較しつつ、図2ないし図4を用いて説明する。

## 【 0 0 2 1 】

図 2 は、従来型のデータベース（エンティティ）間の関係付けを説明する図である。ここでは、エンティティ A が親でエンティティ B が子というデータベースの階層構造を示している。この従来方式では、エンティティ B のデータレコード（インスタンス）中に、親であるエンティティ A を指示するポインタ情報が保持されている。したがって、階層構造が変更されると、このデータベース B 本体を修正する必要があるが生じる。

## 【 0 0 2 2 】

図 3 は、本発明の階層リンク・テーブルを用いた関係付けを説明する図である。ここでは、処理対象であるエンティティ A と B はポインタ情報を持たない階層ノード・データベースとして構築される。両者の関係は、別途ポインタ情報を有する階層リンク・テーブル X の中で定義される。したがって、階層構造が変更されても、データベースそのものを修正する必要はなく、よりサイズの小さい階層リンク・テーブルの内容を修正するだけでよい。

## 【 0 0 2 3 】

図 4 は、本発明の階層リンク・テーブル方式を従来方式と比較しつつ説明する図である。この図では、ある業務処理プログラムが 6 つのエンティティ（実際にはそのインスタンス）からなるデータベースを利用する。その階層構造をみると、エンティティ 1 が最上位ノードであり、その下層の子ノードとしてエンティティ 2 と 3 が、エンティティ 2 の下層にエンティティ 4 が、エンティティ 3 の下層にエンティティ 5 と 6 が位置している。このデータ構造を従来方式で記述すると中欄のようになる。図示されているように、各データレコード（データフィールド）中には、データ本体である属性情報の他に、親子関係を示す（具体的には親エンティティを指示する）ポインタ情報が格納されている。同じデータ構造を階層リンク・テーブルで記述すると下欄のようになる。図示されているように、ポインタ情報は階層リンク・テーブルのデータレコード（データフィールド）にのみ格納され、階層ノード・データベースのデータレコード（データフィールド）には属性情報が保持されるのみである。

## 【 0 0 2 4 】

図 5 は、階層ノード・データベースの内容を説明する図である。階層ノード・データベースは、すべての業務処理プログラムに必要なエンティティのそれぞれにノード ID（識別子）を付与し、各エンティティごとにノード属性としてデータ本体を格納する。すなわち、階層ノード・データベースは、各業務処理プログラムから共通に利用されうるデータを、ノード・データとして各データレコード（データフィールド）中に保持しているわけである。各業務処理プログラムは、前述の階層リンク・テーブルを通じてこのノード ID を参照することによって、必要なデータベースおよびその階層構造を特定することができる。

#### 【 0 0 2 5 】

図 6 は、業務処理プログラムごとに異なるデータ階層構造を有する場合の階層リンク・テーブルの実施例を説明する図である。業務処理プログラムが利用する階層ノード・データベースは、ここでは「エンティティ」として示されている。業務処理プログラム A P 1 において利用されるデータベースは、エンティティ 1 が最上位ノードで、その下層にエンティティ 2、5、4、3 が図示するような関係で連なる階層構造を有している。この A P 1 に関するデータの階層構造は、階層リンク・テーブル T 1 に格納されるエンティティ間の関係を記述するリンク 2 8 ～ 3 1 のリンク情報により保持される。これに対し、業務処理プログラム A P 2 において利用されるデータベースは、エンティティ 6 が最上位ノードで、その下層にエンティティ 2、1、4 が図示するような関係で連なる階層構造を有している。この A P 2 に関するデータの階層構造は、階層リンク・テーブル T 2 に格納されるエンティティ間の関係を記述するリンク 3 2 ～ 3 4 のリンク情報により保持される。この図からも理解できるように、本発明の階層リンク・テーブルは、各業務処理プログラムごとに作成され保持される。

#### 【 0 0 2 6 】

図 7 は、上記のような業務処理プログラムと階層リンク・テーブルと階層ノード・データベースの関係を説明する図である。階層リンク・テーブル A は、業務処理プログラム A に対応して設けられる。階層リンク・テーブル A は、この業務処理プログラム A から見た関連付け情報（階層構造情報）や後述の有効期間情報等を格納しており、業務処理プログラム A は、この階層リンク・テーブル A の内容

に基づいて、処理対象である階層ノード・データベース X、Y、Z にアクセスして必要なデータを取得し、これを用いて特定の業務を処理する。

#### 【0027】

図 8 は、本発明の階層リンク・テーブルを用いたデータベース管理方式を通信事業者の料金業務支援システムに適用した例を説明するための図である。ここでは、各顧客ごとに、図示するような 5 つの業務処理プログラム（A P 1 ～ A P 5）が稼動している。それぞれの業務からみた階層構造はすべて異なるので、それぞれの業務処理プログラムに対応した階層リンク・テーブルを定義される。階層構造の変更は、この階層リンク・テーブルを変更することにより柔軟に対応できる。なお、各顧客ごとに別の階層構造を用いることができる（あるいはその場合が多い）ことにも留意されたい。また、図では省略されているが、A P 1 の契約サービス基本構造内の各商品／サービスは、上位層のカスタマ以外にも他の構造の最下位層にも関連付けられている。これらの業務のうち、いくつかを取り上げて、その実施例を図 9 および図 1 0 を用いて説明する。

#### 【0028】

図 9 は、図 8 のうち割引計算業務（A P 2）における階層構造の実施例を説明する図である。ここでは、階層ノード・データベース「プロダクト・カタログ」のプロダクト I D 欄に示されるとおり、7 つのノードでデータベースの階層が構成されている。なお、同図では、例えば「属性 4」が 2 つ存在しているが、これは参照の便宜上記述されているものであり、同じ内容の情報が格納されていることを意味するものではない。階層ノード・データベースでは、各ノードごとに対応する個々の情報が格納されていることに留意されたい（図 1 0 でも同様）。図 9 で示す割引計算業務プログラム A P 2 の階層構造は、階層リンク・テーブル「割引構造 1」で示される。このテーブルから判断できるように、ここでは「料金計算 4」、「割引 5」、「割引 6」の 3 つの親ノードが存在し、それぞれに対応する子ノードが定義されている。例えば、割引 5 が適用されるのは、商品／サービス 1 と 2 のみに対してであり、商品／サービス 3 や 4 には適用されないことが分かる。

#### 【0029】

図10は、図8のうち請求集計業務（AP4）における階層構造の実施例を説明する図である。基本的には図9と同様の説明が妥当するが、ここでは「組織体DB」という階層ノード・データベースが存在している点異なる。なお、図10の階層ノード・データベース「プロダクト・カタログ」は、説明の便宜のため図9と異なるものとして記述されているが、図9と同一のデータベースであってよい。

#### 【0030】

以上の段落においては、本発明の階層リンク・テーブルを用いたデータベース管理システムについて説明した。以下では、さらに有効な仕組みとして、「有効期間情報」を用いたデータベース管理方式について説明する。

#### 【0031】

図11は、有効期間情報を階層リンク・テーブルや階層ノード・データベースのデータレコード（データフィールド）に設定する実施例を説明する図である。有効期間情報とは、テーブルやデータベース内の各データレコードが、対応する業務処理プログラムからみて有効である期間をいい、通常は、有効になる日時（有効開始日）と有効でなくなる日時（有効終了日）をデータとして記述することにより設定できる。例えば、図中の階層リンク・テーブルへの適用例において、リンク30は1999年1月10日から有効となり同年3月31日まで有効なものとして存続する。この有効期間が終了した後は、親ノード101と子ノード103間の親子関係も終了する。一方、リンク31は1999年4月1日から有効となり現在も有効なものとして存続している。したがって、同年4月1日以降は、親ノード101と子ノード102の親子関係が有効なものとして適用される。

#### 【0032】

図11では、階層ノード・データベースのデータレコードにも有効期間情報を設定している。しかしながら、階層ノード・データベースに有効期間情報を設定するか否かは任意である。階層ノード・データベースは、一般に、各種業務処理プログラムから共通にアクセスされるので、アクセスしてくるすべての業務処理プログラムにとって有効な日時をセットしたり、主要な業務処理プログラムにとって有効な日時を複数個セットしたりする場合には、階層ノード・データベース側



にも有効期間情報を設定する意義がある。例えば、特定日以降からいっせいに新商品の提供をスタートする（あるいはストップする）ような場合である。逆に、特定の業務処理プログラムに対してのみ有効期間を設定したいような場合には、階層リンク・テーブル側の有効期間情報のみを調整すればよく、階層ノード・データベース側でかかる情報を保持する必要はない。

#### 【 0 0 3 3 】

業務処理プログラムは、対応する階層リンク・テーブルの有効期間情報を見て処理を実行するので、当該有効期間外の処理を間違えて実行することはない。例えば、業務処理プログラムが一定の期間単位で処理を行う場合（例えば、一ヶ月間に格納されたレコードを、その月の月末にまとめて集計／計算処理する等）、月の途中でデータベース内のデータレコードとこれに関係付けられた別のデータレコードとの関係／階層構造が変更になる場合がある（例えば、顧客 A がある電話番号を別の人 B に譲渡した場合等のように、いわゆるインスタンス・レベルで親子関係の変更／所属変更等が発生するような場合）。従来方式では、変更の生じた時点で一旦業務処理プログラムを走行させる必要があったが、有効期間情報を用いた本発明の構成によればこのような処理を実行する必要はない。

#### 【 0 0 3 4 】

図 1 2 は、有効期間情報が設定された階層リンク・テーブルを適用した実施例を説明する図である。ここでは、上述のような通信事業者の料金計算／請求集計処理の事例に適用したものとして説明する。この想定事例では、前月の 2 0 日から当月の 1 9 日までの顧客のサービス利用情報（呼情報）に基づき、当月の 2 6 日に料金／割引計算を実行するものとする（ここでは 7 月が当月）。まず、（a）月の途中での変更がなかった場合の料金計算／割引計算について説明する。業務処理プログラムは、必要に応じてサイクル制御表（サイクル・コントロール・テーブル）の指示にしたがって、料金計算／請求集計日である 2 6 日に、6 月 2 0 日から 7 月 1 9 日までの呼情報を選択して料金計算／割引計算を行い、その実行結果を割引 1 の下にリンクする。サイクル制御表とは、ある一連の業務処理が一定の時間間隔（年単位、四半期単位、月単位、週単位、日単位等）で実行されるような場合に、その実行のタイミングと時間間隔（サイクル）を記述しておくた

めのテーブル（データベースの一種）をいう。特に図示はしていないが、このようなサイクル表を用いて、その表中に業務処理プログラムの識別子（プロダクトIDともいう）を指示しておけば、本発明のもとで柔軟に管理できる階層構造を用いて、自動的に必要な業務処理プログラムを実行できる。

## 【0035】

次に、（b）月の途中で変更がなされた場合の料金計算／割引計算について説明する。ここでは7月13日に前サービス（割引1）のもとでの契約が終了し、7月14日から新サービス（割引2）のもとでの契約がはじまるものとする。通常のサイクル制御表からの制御では、6月20日から7月19日までの呼情報を選択して料金計算／割引計算しようとするが、割引1については、電話1／電話2との関係（契約状況）を示す階層リンク・テーブルの有効終了日が7月13日になっているため、6月20日から7月13日までの計算しか行わない。結果は割引1の下にリンクしておく。割引2については、階層リンク・テーブルの有効期間情報で指定された7月14日以降（同19日まで）の期間に対して同様に料金を計算し、結果を割引2の下にリンクしておく。

## 【0036】

また、（c）請求料金集計についてであるが、請求1（請求集計プログラム）は、自己の下層にリンクされている「割引」という階層ノード・データベースのデータレコード（割引1および割引2）を見て、それらにリンクされている計算結果を集計するだけで請求金額を算出し、これをもとに顧客である会社2に対する請求処理を実行することができる。

## 【0037】

図13は、本発明によるデータベース管理方式の柔軟性をより高める可変長コラムの実施例について説明する図である。上述の階層ノード・データベースのすべてまたは主要なものを、固定長コラムと可変長コラムから構成される同一レコード様式のデータベース形式とすることができる。具体的にいうと、各データレコードの固定長コラムには、レコード識別子（ノード識別子）や有効期間情報等、そのデータサイズが常に一定な必須のデータ項目のみを格納し、当該データベース（エンティティ）の属性情報はすべて可変長コラムに格納する。このようにす

ることにより、顧客、サービス、設備等多様なタイプのエンティティのレコード形式を共通化して、ごく少数のスーパータイプに整理統合し、データベースの維持・管理作業の負担を軽減することができる。図 13 では、通信事業者の設備データベースの維持・管理を容易にし、変更への柔軟性を確保するために、設備関連エンティティのタイプや属性情報を全て可変長カラム部分に格納し、固定長カラム部分には有効期間情報のみを格納した例を示しており、これにより多種多様なエンティティを同一のデータベース構造とすることができる。

#### 【0038】

本発明の利点を、いくつかの実施例をまじえて説明してきたが、さらなる利点がある。それは、業務のプロセスごとに互いに論理的に分離・独立した処理対象の制御が可能となる点である。これを、図 8 に示す業務のいくつかを参照しつつ図 14 および図 15 を用いて説明する。

#### 【0039】

図 14 は、本発明のデータベース管理方式を通信事業者の料金業務の一つである割戻し金額計算に適用する実施例を説明する図である。割戻し計算とは、各顧客単位で計算した料金（割引計算を含む）を、顧客が指定する請求先（割戻し先）に顧客が指定する割合等で請求するための計算処理をいう。図 14 では、（a）の料金計算／割引計算階層構造と（b）の請求階層構造とを別の階層リンク・テーブルを定義することにより分離している。オペレーション上は、（a）の料金計算／割引計算階層構造に基づき各種の割引サービス等に応じた料金計算を行ない、（b）の請求階層構造に基づき、顧客が指定する方式で料金を請求する。これにより、例えば（b）で請求先を顧客が指定・変更できるようになる等、変更のための柔軟性が増し、また各階層構造の保守・メンテナンスも容易になる。通常、（a）の料金計算／割引計算階層構造は、商品・サービスの仕様により決まり、（b）の請求階層構造は、顧客の組織・場所等により決まる。この図での割引計算時の割戻しは、単純な比例配分としておき、請求金額集計時に割引前料金集計階層構造とは別に割引額集計階層構造を用意しており、これにより、顧客から要求されるさまざまな割引額配分（割戻し）方法すなわち比例配分（貢献度配分）／同額配分／特別配分等が柔軟に実現できる。

## 【0040】

図15は、本発明のデータベース管理方式を通信事業者の料金業務に適用する別の実施例を説明する図である。この図では、各顧客の請求用構造と報告用構造の定義を、別個の階層リンク・テーブルを用いることにより、任意にかつ相互に独立して行う例を示している。これら請求処理に用いられるデータベース間の関係／階層構造を請求グループ構造とも呼ぶが、この請求グループ構造は、従来は顧客の要求が変わるたびに修正作業が必要になり、関連する多くのデータベースの変更という多大な稼働を要する作業となっていた。しかし、この関係付けを階層リンク・テーブルを用いて定義しておけば、このサイズの小さい階層リンク・テーブルを変更するだけで済むので、変更作業は容易に行なうことができる。すなわち、階層リンク・テーブルにより請求グループ構造を定義する方式を持つ本発明のシステムは、顧客の要件変更に対応できるものといえる。

## 【0041】

ここで、「請求」業務処理プログラムは、顧客であるカスタマaへの請求（請求1）処理として、カスタマa、その下層にある部門x...yで利用されたサービスの料金計算結果を集計し、カスタマaの住所に請求する。「レポート」業務処理プログラムは、当該顧客に対しては、レポート10の処理によりカスタマa、部門x...yの集計結果（および明細）をカスタマaの住所に報告し、部門yに対しては、レポート11の処理により部門yの集計結果（および明細）を、部門yの住所に報告する。

## 【0042】

## 【発明の効果】

以上説明したように、本発明の階層リンク・テーブルを採用したデータベース管理方式によれば、業務内容や関連データ（サービス内容や顧客に関する情報を含む）の追加・変更に対しても柔軟かつ迅速に対処できるデータベース・システムが提供される。また、本発明の有効期間情報の構成をさらに備えることで、処理対象となる期間の途中で上記の追加・変更が生じた場合であっても、過去のデータに溯っての処理が適切かつ迅速、簡便に行なえるようになる。

## 【図面の簡単な説明】

【図 1】 本発明を構成するデータ処理システムを説明するブロック図である。

【図 2】 従来型のデータベース（エンティティ）間の関係付けを説明する図である。

【図 3】 本発明の階層リンク・テーブルを用いた関係付けを説明する図である。

【図 4】 本発明の階層リンク・テーブル方式を従来方式と比較しつつ説明する図である。

【図 5】 階層ノード・データベースの内容を説明する図である。

【図 6】 業務処理プログラムごとに設けた階層リンク・テーブルを説明する図である。

【図 7】 業務処理プログラムと階層リンク・テーブルおよび階層ノード・データベースの関係を説明する図である。

【図 8】 本発明を通信事業者の料金業務支援システムに適用した実施例を説明する図である。

【図 9】 図 8 のうち割引計算業務における階層構造の実施例を説明する図である。

【図 1 0】 図 8 のうち請求集計業務における階層構造の実施例を説明する図である。

【図 1 1】 有効期間情報を階層リンク・テーブル等に設定した実施例を説明する図である。

【図 1 2】 有効期間情報が設定された階層リンク・テーブルの実施例を説明する図である。

【図 1 3】 本発明における可変長コラムの実施例を説明する図である。

【図 1 4】 本発明を通信事業者の割戻し計算業務に適用する実施例を説明する図である。

【図 1 5】 本発明を通信事業者の料金業務に適用する別の実施例を説明する図である。

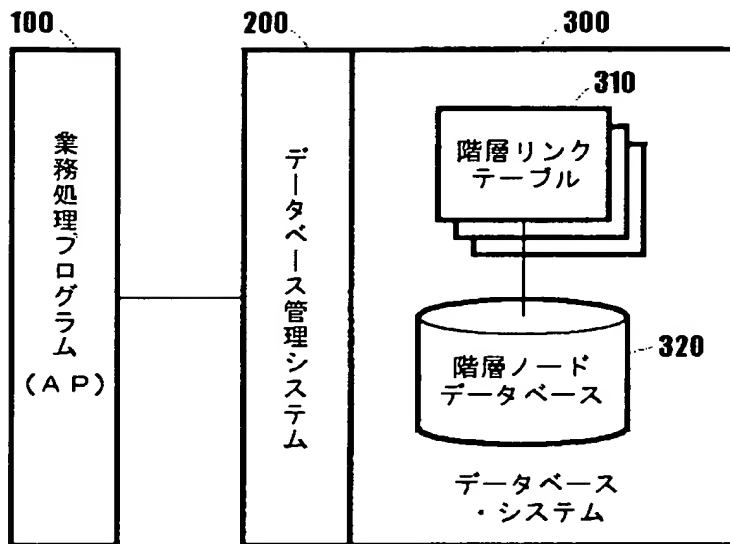
【符号の説明】

1 0 0 . . . 業務処理プログラム、 2 0 0 . . . データベース管理システム、 3 0 0 . . . データベース・システム、 3 1 0 . . . 階層リンク・テーブル、 3 2

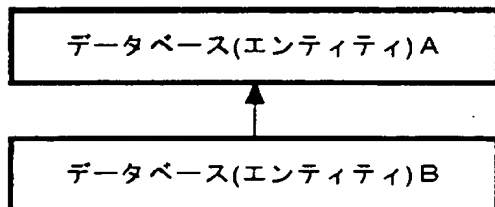
0 . . . 階層ノード・データベース

【書類名】 図面

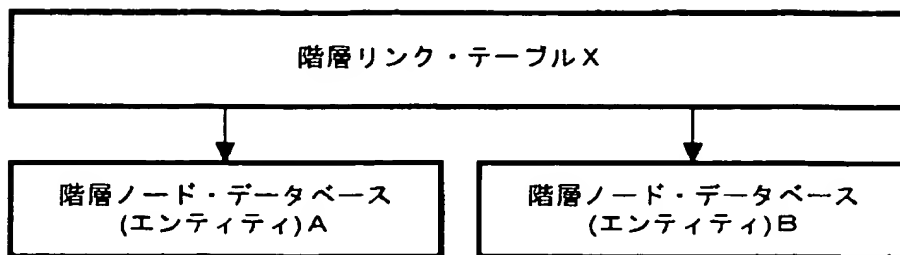
【図 1】



【図 2】

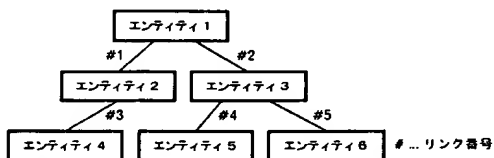


【図 3】



【図 4】

ある業務処理プログラムから見たデータベース間の関係



従来型のデータ格納方法

1		属性
2	親=エンティティ 1	属性
3	親=エンティティ 1	属性
4	親=エンティティ 2	属性
5	親=エンティティ 3	属性
6	親=エンティティ 3	属性

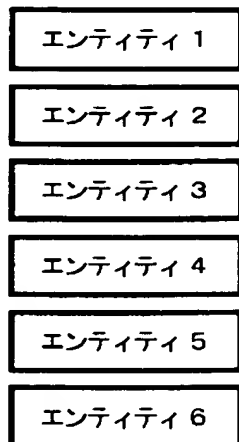
階層リンク・テーブル方式に基づくデータ格納方法

階層ノード・データベース			階層リンク・テーブル		
1		属性	1	親=エンティティ 1	子=エンティティ 2
2		属性	2	親=エンティティ 1	子=エンティティ 3
3		属性	3	親=エンティティ 2	子=エンティティ 4
4		属性	4	親=エンティティ 3	子=エンティティ 5
5		属性	5	親=エンティティ 3	子=エンティティ 6
6		属性			



【図 5】

全業務に必要なエンティティ

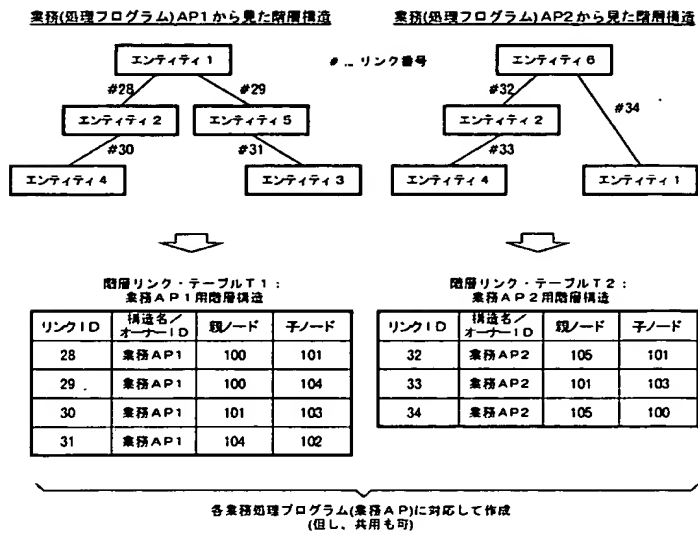


階層ノード・データベース

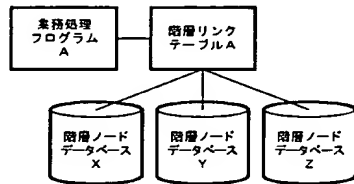
ノードID	ノード属性
100	エンティティID = 1, 名前 = ....
101	エンティティID = 2, 名前 = ....
102	エンティティID = 3, 名前 = ....
103	エンティティID = 4, 名前 = ....
104	エンティティID = 5, 名前 = ....
105	エンティティID = 6, 名前 = ....

各種業務から共通に利用される

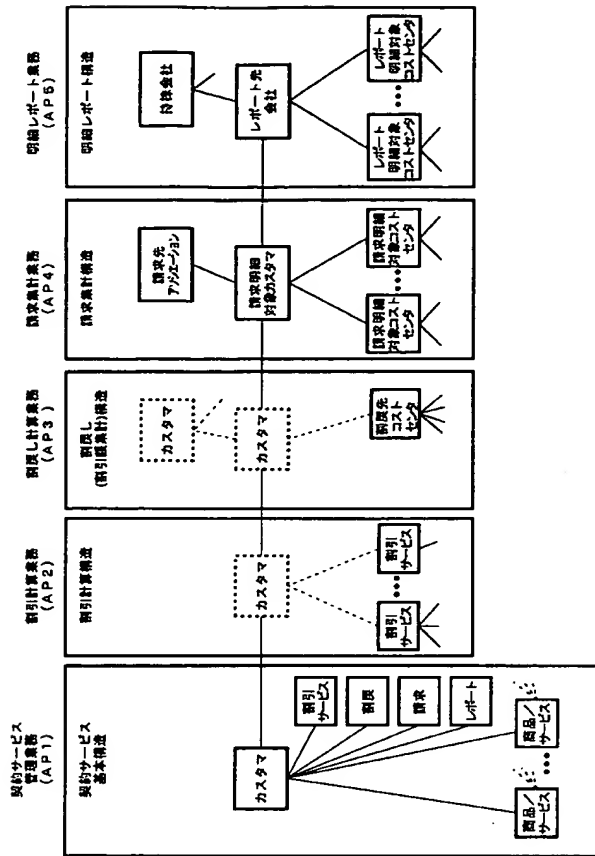
【図 6】



【図 7】

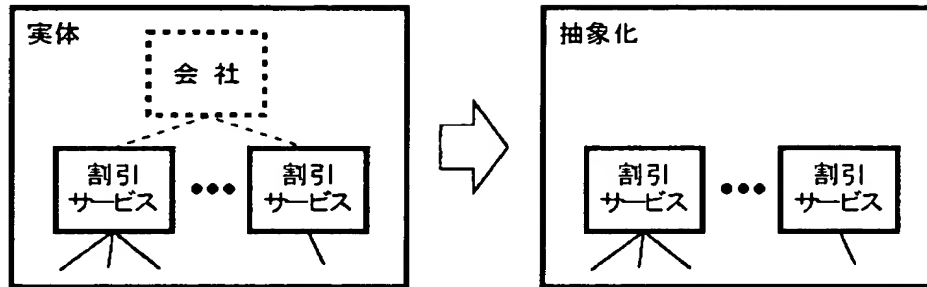


【図 8】



【図 9】

ある顧客の割引計算構造



階層ノード・データベース：プロダクト・カタログ

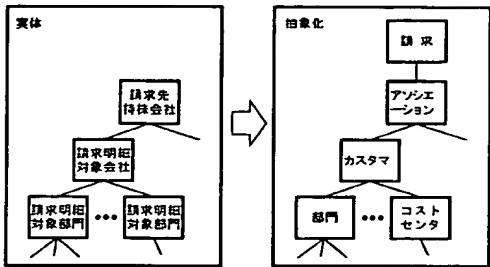
プロダクトID	属 性
商品／サービス 1	属性 1
商品／サービス 2	属性 2
商品／サービス 3	属性 3
商品／サービス 4	属性 4
料金計算 4	属性 4
割引 5	属性 5
割引 6	属性 6

階層リンク・テーブル：割引構造 1

リンクID	オーナー	親ノード	子ノード
リンク 1	割引構造 1	料金計算 4	商品／サービス 1
リンク 2	割引構造 1	料金計算 4	商品／サービス 2
リンク 3	割引構造 1	料金計算 4	商品／サービス 4
リンク 4	割引構造 1	料金計算 4	商品／サービス 3
リンク 6	割引構造 1	割引 5	商品／サービス 1
リンク 7	割引構造 1	割引 5	商品／サービス 2
リンク 8	割引構造 1	割引 6	商品／サービス 1
リンク 9	割引構造 1	割引 6	商品／サービス 3
リンク 10	割引構造 1	割引 6	商品／サービス 4

【図 1 0】

ある顧客の請求集計構造



階層ノード・データベース：総経路DB

組織体ID	属 性
アソシエーション1	属性1
カスタマ1	属性1
部門1	属性1
部門2	属性2

階層ノード・データベース：フロダクト・カタログ

プロダクトID	属 性
請求1	属性1
商品／サービス1	属性1
商品／サービス2	属性2
商品／サービス3	属性3
商品／サービス4	属性4

階層リンク・テーブル：請求構造1

リンクID	オーナー	親ノード	子ノード
リンク1	割引構造1	請求1	アソシエーション1
リンク2	割引構造1	アソシエーション1	カスタマ1
リンク3	割引構造1	カスタマ1	コストセンタ1
リンク4	割引構造1	カスタマ1	コストセンタ2
リンク5	割引構造1	アソシエーション1	カスタマ2

【図 1 1】

階層リンク・テーブルへの適用例：

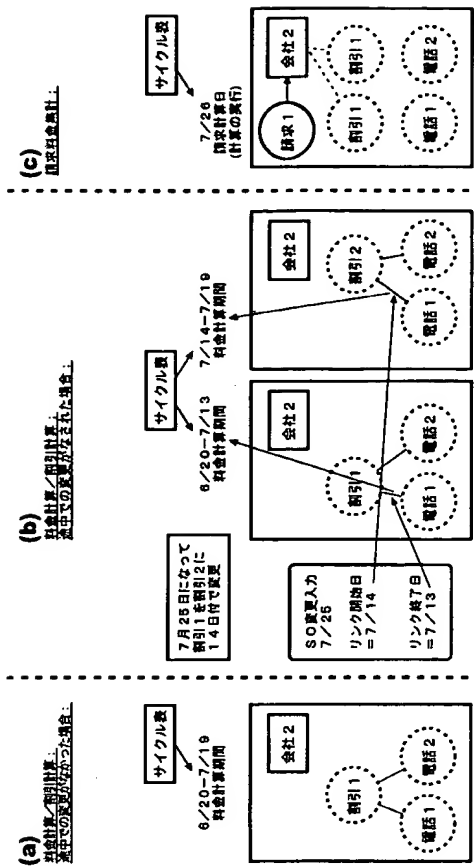
リンクID	有効開始日	有効終了日	構造名／ オーナーID	親ノード	子ノード
28	1999.1.10		業務AP1	100	101
29	1999.1.10		業務AP1	100	104
30	1999.1.10	1999.3.31	業務AP1	101	103
31	1999.4.1		業務AP1	101	102

リンクID	有効開始日	有効終了日	構造名／ オーナーID	親ノード	子ノード
32	1999.1.10		業務AP2	105	101
33	1999.1.10	1999.3.31	業務AP2	101	103
34	1999.1.10		業務AP2	105	100

階層ノード・データベースへの適用例：

ノードID	有効開始日	有効終了日	ノード属性
100	1999.1.1		エンティティID=1, 名前= ....
101	1999.1.1		エンティティID=2, 名前= ....
102	1999.4.1		エンティティID=3, 名前= ....
103	1999.1.1	1999.3.31	エンティティID=4, 名前= ....
104	1999.1.1		エンティティID=5, 名前= ....
105	1999.1.1		エンティティID=6, 名前= ....

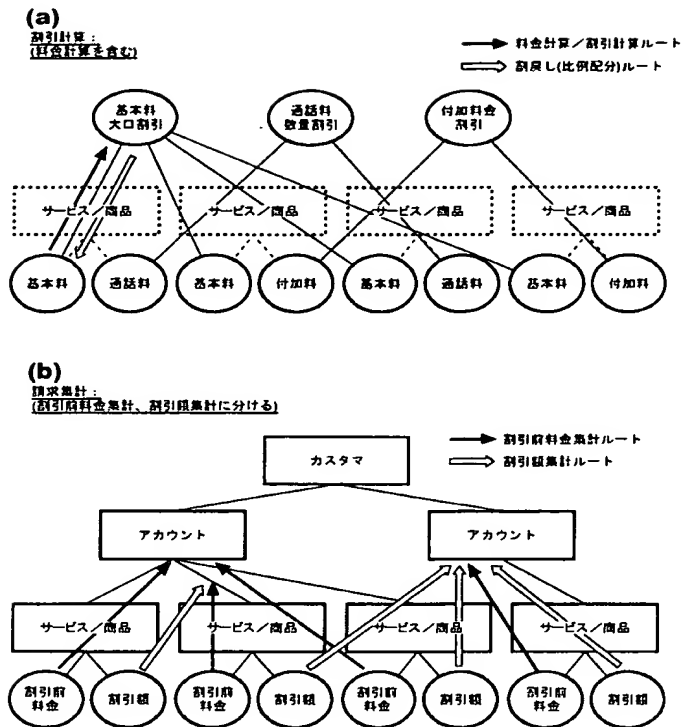
【図 1 2】



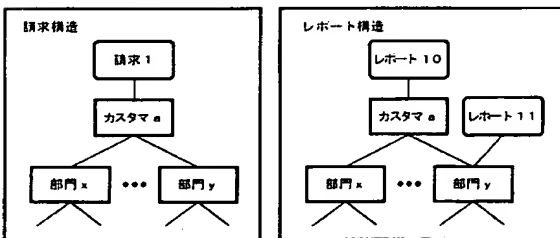
【図 1 3】

ノードID	有効開始日	有効終了日	ノード属性
1999.1.1	1999.3.31		タイプ=最寄住、電柱ID= .....
1999.4.1			タイプ=最寄住、電柱ID= .....
1999.1.1	1999.3.31		タイプ=ケーブル、ケーブルID=、ケーブル名= .....
1999.4.1			タイプ=ケーブル、ケーブルID=、ケーブル名= .....
1999.1.1			タイプ=HH、HH-ID=、HH名= .....
1999.1.1			タイプ=光心線、光心線番号=、ステータス=空き .....
1999.8.1			タイプ=ONU、ONU種別=、ONU-ID= .....
1999.8.1			タイプ=ONU-LC、LCスロット番号= .....
1999.1.1			タイプ=N-SLT、SLT-ID= .....
1999.1.1			タイプ=OSU、OSU収容位置= .....
1999.1.1			タイプ=VCN、VCN収容位置= .....
1999.1.1			タイプ=接続端子かん、接続端子かん番号= .....
1999.1.1			タイプ=メタル心線、心線番号= .....
1999.1.1			タイプ=LXM収容位置、LXM収容位置番号= .....
1999.1.1			タイプ=DSU、DSU-ID= .....
1999.1.1			タイプ=接続端子かん、接続端子かん番号= .....

【図 1 4】



【図 1 5】





【書類名】 要約書

【要約】

【課題】

業務内容や関連データ（サービス内容や顧客に関する情報を含む）の追加・変更に対しても柔軟かつ迅速に対処できるデータベース・システムを提供すること。

【解決手段】

本発明においては、各業務処理プログラムが処理対象とするデータを含んだデータベース中にはデータベース（エンティティ）間の関連付け情報（ポインタ等）を原則として持たせない（階層ノード・データベース）。その一方で、これとは別個にデータベース間の関連付け情報（ポインタ等）を主として有するテーブル（階層リンク・テーブル）を各業務処理プログラムに対応して作成する。各業務処理プログラムは、対応する階層リンク・テーブルを参照することにより、自身が必要とする階層ノード・データベースにアクセスすることができる。新しい要件に基づく業務の追加などデータの階層構造に変更が必要な場合が生じたときは、これに対応する修正を階層リンク・テーブルに反映するだけでよい。また、必要に応じて、有効期間情報をこれらのテーブル等に持たせることができる。これにより、処理対象となる期間の途中で上記の変更が必要になった場合であっても、過去のデータに溯っての処理が適切かつ迅速、簡便に行なえるようになる。

【選択図】

図 1

認定・付加情報

特許出願の番号	平成 1 1 年	特許願	第 3 7 5 7 1 8 号
受付番号	5 9 9 0 1 2 8 6 0 7 4		
書類名	特許願		
担当官	第七担当上席	0 0 9 6	
作成日	平成 1 2 年	1 月	5 日

<認定情報・付加情報>

【提出日】	平成11年12月28日
-------	-------------

出 願 人 履 歴 情 報

識別番号 [592073101]

1. 変更年月日	1992年 4月 3日
[変更理由]	新規登録
住 所	東京都港区六本木3丁目2番12号
氏 名	日本アイ・ピー・エム株式会社